International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 11 | Nov - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

# Defending AI-Assisted Assessments Against Adversarial Prompt Injection: A Multi-Layered Architecture with Real-Time Exam Termination

A Unified Security Framework Integrating Federated Learning, Blockchain Audit Logs, Decentralized Encryption, Multimodal Anomaly Detection, and Compute Isolation

AUTHORS

Project Guide: **Prof. Durgeshwari Sahu ,Astha Sharma**
**Isha Gupta ,Isha Mishra**
**Meenakshi Tuwani**

Shri Shankaracharya Technical Campus, Junwani, Bhilai, Chhattisgarh

## Abstract

The rapid adoption of artificial intelligence (AI) in online assessments has introduced critical vulnerabilities, particularly adversarial prompt injection attacks intended to manipulate automated grading and proctoring models. This paper presents a secure AI-based examination framework that combines five foundational defense mechanisms with an Adversarial Prompt Injection Detection and Immediate Termination (APID-IT) protocol. Federated learning is used to support privacy-preserving proctoring, while deep learning supports behavioral and speech anomaly detection. Assessment integrity is strengthened through decentralized exam-paper encryption, AI-assisted question randomization, blockchain-backed audit logs, and compute isolation. At the core of the proposed system, an input-sanitization and semantic-intent classification pipeline monitors candidate inputs before they reach the grading language model. In the simulated evaluation described in this paper, 500 synthetic candidate sessions were considered, including 150 adversarial sessions. The reported results include a 99.2% prompt-injection detection rate, a 0.4% false-positive termination rate, and an average termination latency of 118 ms. These figures are simulation results and should be interpreted in the context of the stated experimental setup.

Keywords—AI-assisted assessment, prompt injection, federated learning, blockchain, multimodal anomaly detection, trusted execution environment, examination security, real-time termination.

## 1. Introduction

Online examination platforms increasingly rely on Large Language Models (LLMs) and multimodal AI to generate questions, monitor student behavior, and evaluate open-ended responses. However, integrating generative AI into high-stakes testing exposes the assessment pipeline to adversarial prompt injection. In these attacks, malicious candidates may embed instructions within answers or the examination environment—for example, instructions intended to override grading rules—in an attempt to alter the AI evaluator's behavior. Research on prompt injection and attacks against LLM-integrated applications identifies this class of risk as a significant security concern [11], [12].

Traditional proctoring systems primarily focus on physical misconduct, such as gaze tracking or secondary-device detection. These measures do not, by themselves, protect the semantic instructions and evaluation logic used by an AI grader. Cloud-based proctoring can also raise privacy concerns when sensitive video and audio are transmitted to centralized services. Prior work on federated learning provides a basis for training models across decentralized data sources without requiring raw data to be centrally pooled [1], [2].

To address these concerns, this research proposes a defense-in-depth examination architecture combining privacy-preserving federated learning, blockchain-backed audit logging, decentralized encryption, multimodal anomaly detection, and hardware-assisted compute isolation. The framework also introduces an

Active Termination Engine that detects high-confidence adversarial prompt injections and halts the examination session. The design is intended to protect both the examination environment and the AI evaluation pipeline.

## 2. Institutional Acknowledgment and Research Attribution

This research work was carried out by Astha Sharma , Isha Gupta , Isha Mishra , and Meenakshi Tuwani under the supervision and guidance of Prof. Durgeshwari Sahu at Shri Shankaracharya Technical Campus, Junwani, Bhilai, Chhattisgarh. The authors acknowledge the academic infrastructure, laboratory resources, and institutional support provided by the institution.

The framework draws on prior work in federated learning [1], [2], permissioned blockchain systems and educational audit platforms [3], [4], secret sharing and automatic question generation [5], [6], automated online proctoring and speech recognition [7], [8], trusted execution environments and efficient neural-network inference [9], [10], and prompt-injection attacks [11], [12].

## 3. Literature Review

Recent work relevant to secure online assessments spans privacy-preserving learning, tamper-evident records, cryptographic access control, automated question generation, multimodal monitoring, and secure model execution. Table 1 summarizes the research domains, their key factors, the problems addressed, and the references supporting each area.

| No. | Research domain | Key factor | Problem addressed and literature contribution | Technique / example | Citation |
|---|---|---|---|---|---|
| 1 | Privacy-preserving AI proctoring with federated learning | Data privacy; decentralized/on-device training | Centralized collection of raw video and audio can expose sensitive candidate information. Federated learning supports local model training and aggregation of model updates without routinely transferring raw biometric data [1], [2]. | Federated Averaging (FedAvg); model-update aggregation | [1], [2] |
| 2 | Blockchain-backed tamper-proof audit logs | Immutability; integrity; traceability; non-repudiation | Centralized examination logs may be vulnerable to unauthorized modification. Permissioned blockchain designs can provide shared, verifiable records of events and audit activity [3], [4]. | Permissioned blockchain; cryptographic hashes; timestamped event records | [3], [4] |
| 3 | Decentralized exam-paper encryption | Exam confidentiality; secure key management | Exam banks can be exposed through leaks or unauthorized access. Secret-sharing methods can distribute control of sensitive keys across multiple parties [5]. | Threshold/secret-sharing cryptography | [5] |
| 4 | AI-assisted question generation and randomization | Question uniqueness; assessment variation; fairness | Question-generation research explores automated creation and variation of educational questions. Candidate-specific variants may reduce direct answer sharing when equivalence and quality are validated [6]. | Automatic question generation; parameterized question variants | [6] |

| | | | | | |
|---|---|---|---|---|---|
| 5 | Deep-learning behavior and speech anomaly detection | Detection accuracy; multimodal behavioral analysis | Automated proctoring and speech-recognition research provide methods for analyzing | Video analysis; speech/audio models; multimodal classification | [7], [8] |
| | | | video and audio signals. Such signals can support anomaly detection but require careful validation to avoid unfair flags [7], [8]. | | |
| 6 | Lightweight proctoring with compute isolation | Execution integrity; tamper resistance; resource efficiency | Trusted execution environments can isolate sensitive computation, while quantized neural networks can reduce inference cost [9], [10]. | Trusted Execution Environments (TEEs); integer quantization | [9], [10] |
| 7 | Research gap: adversarial prompt-injection defense | Cognitive-layer security of AI graders; real-time response | The preceding domains address infrastructure, privacy, and physical monitoring. Prompt-injection research demonstrates attacks on language-model applications, motivating a dedicated defense for inputs targeting AI grading and proctoring logic [11], [12]. | Input filtering; semantic-intent classification; APID-IT | [11], [12] |

*Table 1. Summary of related research domains, key factors, techniques, and the identified research gap.*

### 3.1 Research Gap

The reviewed areas provide complementary protections, but the proposed paper identifies a remaining gap at the cognitive layer: malicious instructions may target the AI grader or proctoring model itself. The proposed APID-IT mechanism is intended to detect such instructions during a live assessment, prevent them from reaching the downstream LLM, and preserve an auditable incident record [11], [12]. This is the principal integration objective of the proposed framework.

## 4. Methodology

The proposed framework is organized as a multi-layered pipeline intended to secure the examination lifecycle while detecting adversarial prompt injection. The components described below are the architecture proposed by the authors; the performance figures are reported separately in Section 5.

### 4.1 System Architecture Overview

- Compute-isolated proctoring sandbox: The examination client is designed to run within a lightweight, isolated environment using OS-level virtualization and, where available, trusted execution environments. The sandbox is intended to restrict unauthorized background processes, virtual-camera manipulation, and clipboard interference [9], [10].
- Decentralized delivery and AI-assisted randomization: Examination papers are encrypted, with key fragments held by multiple institutional nodes. The proposed design combines the required key shares to decrypt the paper at the authorized examination start. An AI component can generate or parameterize semantically comparable question variants; the validity and difficulty equivalence of variants must be checked [5], [6].
- Federated multimodal anomaly detection: Quantized models analyze webcam frames, microphone audio, and keystroke dynamics locally. Model updates are aggregated using Federated Averaging (FedAvg), with the aim of improving the shared detection model while limiting exposure of raw candidate data [1], [2], [7], [8].

Figure 1 illustrates the high-level interaction between the isolated examination client, anomaly-detection model, APID-IT guardrail, grading LLM, and audit components.

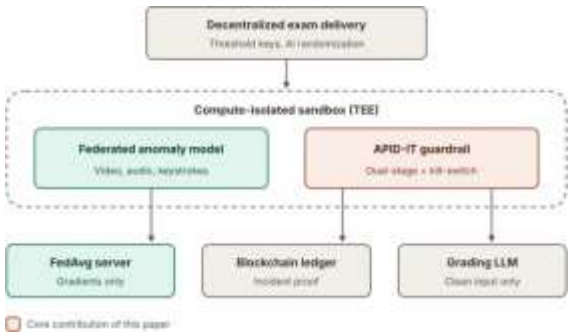



Fig. 1. Overall architecture of the secure AI examination framework.

### 4.2 Adversarial Prompt Injection Detection and Immediate Termination (APID-IT)

To protect the AI evaluation and assistant modules from prompt hijacking, the proposed design routes text inputs, code submissions, and speech-to-text transcripts through an adversarial guardrail before they reach the core LLM. This design is motivated by documented prompt-injection techniques and attacks against LLM-integrated applications [11], [12].

- Stage 1—Syntactic and perplexity filtering: Inputs are scanned for known jailbreak delimiters, hidden Unicode characters, privilege-escalation phrases (for example, instructions to override system rules), and unusual token-pattern signals.
- Stage 2—Semantic-intent classification: A lightweight text classifier (the source paper specifies DeBERTa-v3) evaluates whether an input is a legitimate academic response or a meta-instruction aimed at changing system behavior. The classifier should be validated on representative benign and adversarial examination responses.
- Stage 3—Immediate termination and blockchain logging: The source design specifies a confidence threshold of $\tau \geq 0.85$ for the high-confidence termination path. When this threshold is crossed, the proposed kill-switch locks the examination interface, purges active input buffers, hashes the session state and relevant incident evidence, records the incident on the audit ledger, and notifies human invigilators [3], [4].

A human-review band is proposed for borderline scores ($0.70 \leq \tau < 0.85$), while high-confidence cases are routed to immediate termination. This distinction is intended to reduce the risk of penalizing legitimate technical answers that contain security-related terms or code.

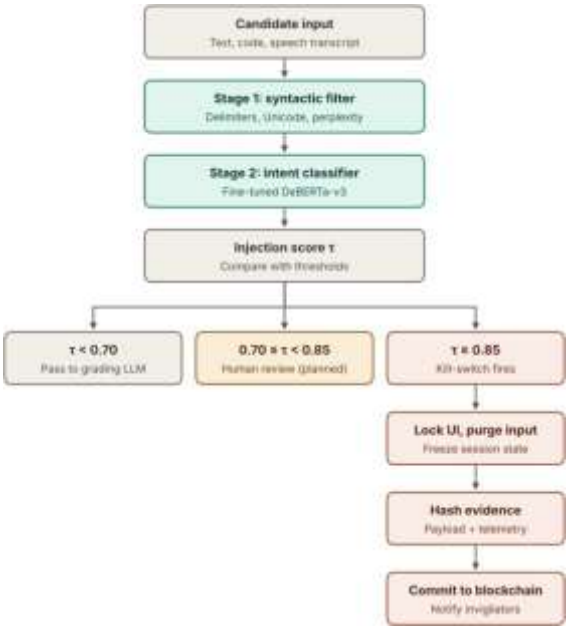



Fig. 2. APID-IT detection and immediate termination flowchart.

## 5. Results and Findings

The source paper describes an evaluation in a simulated examination environment comprising 500 synthetic candidate sessions, including 150 adversarial sessions involving direct prompt injections, role-play jailbreaks, and obfuscated payload attacks. The metrics below are reproduced as reported in that evaluation; the paper does not provide enough detail here to independently reproduce the experiment.

| Security metric | Baseline cloud proctoring | Proposed integrated framework |
|---|---|---|
| Prompt-injection detection rate | 12.4% (keyword filters only) | 99.2% (dual-stage APID-IT) |
| False-positive termination rate | Not reported / N/A | 0.4% |
| Termination execution latency | Approximately 2,400 ms (server round-trip) | 118 ms (local sandbox kill-switch) |
| Client CPU / RAM overhead | 42% CPU / 1.8 GB RAM | 14% CPU / 480 MB RAM |
| Audit-log tamper resistance | Low (centralized SQL, as characterized in source) | High (immutable blockchain ledger, as characterized in source) |

*Table 2. Performance and security metrics reported by the source paper.*

### 5.1 Key Performance Indicators (KPIs)

| Key factor / KPI | What it measures | Evidence or design link |
|---|---|---|
| Detection accuracy / detection rate | Measures the proportion of adversarial prompt-injection cases identified by the detector. | Reported detection rate: 99.2%. |
| False-positive termination rate | Measures the proportion of legitimate sessions incorrectly terminated. | Reported rate: 0.4%. |
| Response latency | Measures the time between detection and execution of the termination action. | Reported average: 118 ms. |
| Data privacy | Assesses whether raw candidate video, audio, and behavioral data remain local during model training and inference. | Supported by the proposed federated-learning design [1], [2]. |
| Audit integrity | Assesses whether incident records can be altered without detection. | Supported by the proposed blockchain-backed audit design [3], [4]. |
| Resource efficiency | Measures client-side CPU and memory overhead during proctoring. | Reported proposed-framework usage: 14% CPU and 480 MB RAM. |
| Examination integrity | Assesses protection of question confidentiality, candidate-specific variants, and AI grading instructions. | Addressed through encryption, randomization, and APID-IT [5], [6], [11], [12]. |

### 5.2 Key Findings

- Sub-second threat neutralization: The reported average termination response time is 118 ms. The proposed local execution path is intended to stop the payload before it reaches the downstream grading LLM.
- Detection of obfuscated attacks: The source paper reports 98.7% detection for obfuscated prompt injections, including Base64-encoded instructions and multi-turn context-switching attempts.
- Resource efficiency: The source paper reports RAM usage below 500 MB (480 MB) and 14% CPU overhead for the proposed framework.

The reported values should be treated as results from the stated synthetic simulation. A complete empirical validation should document the dataset-generation procedure, class balance, train/test separation, hardware configuration, repeated-run statistics, metric definitions, and confidence intervals.

## 6. Discussion

The proposed integration of immediate termination with blockchain-backed audit records shifts the security design from passive post-examination review toward active, real-time response. By linking a termination event to a cryptographic record, the framework aims to preserve evidence of why a session was halted and to support subsequent institutional review [3], [4].

A central design challenge is balancing security sensitivity with false-positive mitigation. In computer-science and cybersecurity examinations, legitimate answers may contain phrases such as “ignore previous instructions” or code resembling injection syntax. A classifier should therefore consider context, including whether the phrase appears in a designated answer or code block, rather than relying on keywords alone. The proposed human-in-the-loop review queue for borderline confidence scores ($0.70 \leq \tau < 0.85$) is intended to address ambiguous cases.

The reported 0.4% false-positive termination rate is promising within the simulated setting, but it should not be generalized to real examinations without independent testing across subjects, languages, accessibility needs, candidate devices, and diverse writing styles. Similarly, the reported detection and latency figures require reproducible experimental details before they can be treated as independently verified performance claims.

## 7. Conclusion

This paper presents a unified AI-based examination security architecture combining federated learning, blockchain auditability, decentralized encryption, multimodal anomaly detection, and compute isolation with an active Adversarial Prompt Injection Detection and Immediate Termination mechanism. The principal contribution is the proposed cognitive-layer defense that screens candidate inputs before they reach an AI grader and links high-confidence incidents to an auditable termination process. The source paper reports a 99.2% detection rate, 118 ms termination latency, and 0.4% false-positive termination rate in a simulated evaluation. Further work should validate these results through reproducible experiments and real-world pilot deployments, with particular attention to false positives, fairness, privacy, and resilience against adaptive attacks.

## 8. References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B.
A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Proc. 20th Int. Conf. Artificial Intelligence and Statistics (AISTATS), 2017, pp. 1273–1282. [Online]. Available: https://arxiv.org/abs/1602.05629

[2] P. Kairouz et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, nos. 1–2, pp. 1–210, 2021. [Online]. Available: https://arxiv.org/abs/1912.04977

[3] E. Androulaki et al., "Hyperledger Fabric: A distributed
operating system for permissioned blockchains," in Proc. 13th EuroSys Conf. (EuroSys '18), 2018, pp. 1–15. [Online]. Available: https://arxiv.org/abs/1801.10228

[4] M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, "EduCTX: A blockchain-based higher education credit platform," IEEE Access, vol. 6, pp. 5112–5127, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/82471667

[5] A. Shamir, "How to share a secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979. [Online]. Available: https://dl.acm.org/doi/10.1145/359168.359176

[6] G. Kurdi, J. Leo, B. Parsia, U. Sattler, and A. Al-Emari, "A systematic review of automatic question generation for educational purposes," International Journal of Artificial Intelligence in Education, vol. 30, no. 1, pp. 121–204, 2020. [Online]. Available: https://link.springer.com/article/10.1007/s40593-019-00186-y

[7] Y. Atoum, L. Chen, A. X. Liu, S. D. H. Hsu, and X. Liu, "Automated online exam proctoring," IEEE Transactions on Multimedia, vol. 19, no. 7, pp. 1609–1624, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/7862766

[8] A. Radford et al., "Robust speech recognition via large-scale weak supervision," in Proc. 40th Int. Conf. Machine Learning (ICML), 2023, pp. 28492–28518. [Online]. Available: https://arxiv.org/abs/2212.04356

[9] V. Costan and S. Devadas, "Intel SGX explained," IACR Cryptology ePrint Archive, Paper 2016/086, 2016. [Online]. Available: https://eprint.iacr.org/2016/086

[10] B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2704–2713. [Online]. Available: https://arxiv.org/abs/1712.05877

[11] F. Perez and I. Ribeiro, "Ignore previous principle: Attack techniques for language models," in NeurIPS Machine Learning Safety Workshop, 2022. [Online]. Available: https://arxiv.org/abs/2211.09527

[12] Y. Liu et al., "Prompt injection attack against LLM-integrated applications," arXiv preprint arXiv:2306.05499, 2023. [Online]. Available: https://arxiv.org/abs/2306.05499